

Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc Networks.

EPFL Technical Report IC/2003/31

Sonja Buchegger, Jean-Yves Le Boudec

*EPFL-IC-LCA
CH-1015 Lausanne, Switzerland*

Abstract

Some misbehavior detection and reputation systems in mobile ad-hoc networks rely on the dissemination of information of observed behavior, which makes them vulnerable to false accusations. This vulnerability could be removed by forbidding the dissemination of information on observed behavior in the first place, but, as we show here, this has more drawbacks than a solution that allows dissemination and copes with false accusations. We propose a method for reducing the impact of false accusations. In our approach, nodes collect first-hand information about the behavior of other nodes by direct observation. In addition, nodes maintain a rating about every other node that they care about, in the form of a continuous variable per node. From time to time nodes exchange their first-hand information with others, but, using the Bayesian approach we designed and present in this paper, only second-hand information that is not incompatible with the current rating is accepted. Ratings are slightly modified by accepted information. The reputation of a given node is the collection of ratings maintained by others about this node. By means of simulation we evaluated the robustness of our approach against several types of adversaries that spread false information, and its efficiency at detecting malicious nodes. The simulation results indicate that our system largely reduces the impact of false accusations, while still benefiting from the accelerated detection of malicious nodes provided by second-hand information. We also found that when information dissemination is not used, the time until malicious nodes are detected can be unacceptable.

Key words: Mobile ad-hoc networks, misbehavior detection, reputation systems, Bayesian statistics

1 Introduction

The fast detection of malicious nodes is vital in mobile ad-hoc networks, since they rely on the cooperation of nodes for routing and forwarding, and misbehavior can seriously degrade the performance and jeopardize the functionality of network [11, 13, 4]. Misbehavior detection and reputation systems provide an incentive to cooperate in such infrastructure-less networks. In addition, they mitigate the effect of misbehavior by isolating malicious nodes.

In previous work we introduced such a distributed misbehavior detection and reputation system, called CONFIDANT [4], where nodes warn each other about malicious nodes. Although it enables the isolation of malicious nodes, it is vulnerable to false accusations, if trusted nodes lie or if several liars collude.

There are several simple ways to deal with this vulnerability. One could rely exclusively on first-hand information and not allow the dissemination of information at all. This solution can be prohibitively slow, as the nodes have to wait until they have a bad experience until they can conclude that another node misbehaves. Another approach would be to allow only the dissemination of positive reputation information. False accusations are not an issue in positive reputation systems, since no negative information is kept [10, 7], however, the disseminated information could still be false praise and result in a good reputation for malicious nodes. Moreover, even if the disseminated information is correct, one cannot distinguish between a malicious node and a new node that just joined the network. Many reputation systems build on positive reputation only [18], some couple privileges to accumulated good reputation, e.g. for exchange of gaming items or auctioning [17]. Positive reputation systems are thus used for where one has a choice of transaction partners and wishes to find the best one. In mobile ad-hoc networks, the requirements are different, the focus is on the isolation of malicious nodes.

We deem the combined use of both positive and negative reputation adequate for the context of mobile ad-hoc networks, as we are interested in the cooperation factor calculated as the frequency of misbehavior *relative* to the total activity of a node in a network. Moreover, the nature of the disseminated information should match the nature of first-hand information or experiences. If a node keeps track of both positive and negative behavior of other nodes, the disseminated information considered should reflect the same kind of knowledge in order not to introduce a bias in either direction.

Email addresses: sonja.buchegger@epfl.ch (Sonja Buchegger),
jean-yves.leboudec@epfl.ch (Jean-Yves Le Boudec).
URLs: <http://www.icapeople.epfl.ch/sbuchegg> (Sonja Buchegger),
<http://lcawww.epfl.ch/leboudec> (Jean-Yves Le Boudec).

The main properties of a reputation system are the representation of reputation, how the reputation is built and updated, and for the latter, how the ratings of others are considered and integrated. The reputation of a given node is the collection of ratings maintained by others about this node. In our approach, nodes maintain a rating about every other node that they care about, in the form of a continuous variable per node. We represent the rating that $node_i$ has about $node_j$ as a function $R_{i,j}$ of α and β , which are the number of malicious and regular behavior instances, respectively. At each direct observation of behavior, the rating is updated accordingly. To take advantage of disseminated information, i.e., to learn from observations made by others before having to learn by own experience, we need a means of incorporating the reputation ratings into the view of an individual node. To this end, from time to time nodes exchange their first-hand information with others, but, using the Bayesian approach we designed and present in this paper, only second-hand information that is not incompatible with the current rating is accepted. By incompatible we mean that $R_{k,j}$, the rating of $node_k$ about $node_j$, deviates too much from $R_{i,j}$ for $node_i$ to consider it. If, however, the second-hand information received is compatible, it is accepted and slightly modifies $R_{i,j}$. In the particular case of misbehavior detection in mobile ad-hoc networks we want to give the most emphasis on reputation built by actually observed behavior, second-hand information should obtain less weight, since a node trusts its own observations more than a report from a random other node.

In a mobile ad-hoc network, the point of keeping rating records about other nodes of the network is to be able to make more informed decisions about whether to forward for another node, which path to choose, whether to avoid another node and delete it from the path cache, and whether to warn others about another node. Using the Bayesian approach, decisions can be made minimizing the risk for a loss, e.g., minimizing the risk of wrong classification of events, of deeming another node malicious, although it is not, or, vice versa, the risk of not recognizing a node as malicious although it actually misbehaves. If the rating of a node in the table has deteriorated so much as to fall out of a tolerable range, the suspect node is declared “detected” and some action can be triggered.

In this paper, we apply the Bayesian approach to reputation updates, however, it can also serve for event classification of observations, i.e., whether they are regular protocol events or malicious attacks, as well as for trust classification to evaluate nodes according to their cooperation in the reputation system itself independent from their cooperation in the routing and forwarding according to the protocol. Dynamic trust adaptation according to the compatibility metric given by the rating deviation could be considered. However, we use the simpler approach of not discriminating between nodes and thus treating each received information on a case-by-case basis and evaluate its utility solely on

the grounds of how much it deviates from the rating the recipient already has. Trust management is thus rendered obsolete in this particular approach.

In our evaluation, we consider several types of adversaries. In general, adversarial nodes can act maliciously on two different levels. They can misbehave in routing or forwarding, i.e. in the normal operation of an ad-hoc network. We call this the network capabilities of the adversary, and we call a node engaged in network misbehavior a malicious node. Additionally they can try to exploit the incentive mechanism at the meta-level, i.e. in the case of a reputation system they can lie about the reputation of others in order to obtain some benefit or to have other nodes isolated. We call this the reputation capabilities and refer to the adversary as liar.

The liar is interested in destabilizing the reputation system by spreading false information. We consider the following ways of achieving that.

- Reverse the parameters α and β before showing them to another node. This way, a plausible distribution is retained, innocent nodes are punished, and malicious nodes are rewarded.
- Only slightly worsen reputation according to received reputation distribution, by increasing α and decreasing β . A liar could try to create instabilities by lying only so much as to not be discarded as incompatible, yet sufficiently to worsen the reputation of another node gradually over time.
- Improve reputation of another malicious node by slightly decreasing its α and increasing its β .

We assume sufficient identity persistence for a reputation system to work, i.e. that nodes cannot change their identity too easily. This can partly be achieved by using cryptographically generated identities that prevent impersonation [14]. For the creation of new identities, expensive pseudonyms could be used.

By means of simulation we evaluated the robustness of our approach against the types of liars described above, and its efficiency at detecting malicious nodes.

The simulation results indicate that our system largely reduces the impact of false accusations, while still benefiting from the accelerated detection of malicious nodes provided by second-hand information. We also found that when information dissemination is not used, the time until malicious nodes are detected can be unacceptable.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. Our Bayesian solution proposal is detailed in Section 3 and its performance evaluation follows in Section 4. Section 5 offers a discussion and future directions, and Section 6 concludes the paper.

2 Related Work: Misbehavior Detection and Reputation Systems

In the following we describe and discuss several misbehavior detection and reputation systems that are fully distributed and hence potential solutions for mobile ad-hoc networks or peer-to-peer networks. For each of these we describe the strategy used to detect malicious nodes and to cope with false accusations and relate it to ours.

The following protocols either rely only on first-hand information or on positive second-hand information. Since in this paper we evaluate the use of disseminated information, we provide a quantitative reason, namely the speed-up of detection time, why they could potentially benefit from our Bayesian approach while still being robust against false accusations.

Watchdog and path rater components to mitigate routing misbehavior have been proposed by Marti, Giuli, Lai and Baker [11]. They observed increased throughput in mobile ad-hoc networks by complementing DSR with a *watchdog* for detection of denied packet forwarding and a *path rater* for trust management and routing policy rating every path used, which enable nodes to avoid malicious nodes in their routes as a reaction. The nodes rely on their own watchdog exclusively and do not exchange reputation information with others. They thus chose the approach of not using information dissemination, trading off the robustness against longer detection delay.

CORE, a collaborative reputation mechanism proposed by Michiardi and Molva [12], also has a *watchdog* component; however it is complemented by a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request. Nodes only exchange positive reputation information, thus making the same trade-off between robustness against lies and detection speed as the watchdog and path rater scheme, but in addition, false praise can make malicious nodes harder to detect. A performance analysis by simulation is stated for future work.

The protocols discussed next already use negative second-hand information and cope with false accusations by requiring the disseminated information to come from several sources. Our approach could be beneficial for them in the case of collusion of several liars. As opposed to the protocols previously

discussed in this section, the benefit is not straightforward to quantify and thus outside of the scope of this paper.

CONFIDANT (see our papers [4, 3]) stands for ‘Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks’ and it detects malicious nodes by means of observation or reports about several types of attacks and thus allows nodes to route around malicious nodes and to isolate them from the network. Nodes have a *monitor* for observations, *reputation records* for first-hand and trusted second-hand information, *trust records* to control trust given to received warnings, and a *path manager* for nodes to adapt their behavior according to reputation. Simulations for “no forwarding” have shown that CONFIDANT can cope well even with half of the network population acting maliciously. The protocol uses also second-hand information, in the form of warnings, thus negative information only. The problem of false accusations can arise in two cases: when trusted nodes lie or when enough liars collude.

A reputation-based trust management has been introduced by Aberer and Despotovic in the context of peer-to-peer systems [1], using the data provided by a decentralized storage method (P-Grid) as a basis for a data-mining analysis to assess the probability that an agent will cheat in the future given the information of past transactions. The disseminated information is exclusively negative, in the form of complaints that are then redundantly stored at different agents. When agents want to assess the trustworthiness of other agents, they query several agents for complaints about the agent in question. To assess the trustworthiness of the agents responding to the query and thus to avoid relying on lies, a complaint query about that agent can be made. To avoid the exploration of the whole network, the trustworthiness of the responders is said to be given when a sufficient number of replicas returns the same result. An assumption is that the underlying communication network is sound in that the complaints do not have to be routed through malicious nodes, so the approach is not readily applicable to mobile ad-hoc networks.

A context-aware inference mechanism has been proposed by Paul and Westhoff [15], where accusations are related to the context of a unique route discovery process and a stipulated time period. The rating of nodes is based on accusations of others, whereby a number of accusations pointing to a single attack, the approximate knowledge of the topology, and context-aware inference are claimed to enable a node to rate an accused node without doubt. An accusation has to come from several nodes, otherwise the only node making the accusation is itself accused of misbehavior. While this mechanism discourages false accusations, it potentially also discourages correct accusations for fear of being the only denouncer, resulting in reduced information dissemination.

As opposed to the **Byzantine Generals problem**, the nodes in a misbehavior detection and reputation system for mobile ad-hoc networks do not

have to reach a consensus on which nodes misbehave. Each node can keep its own rating of the network denoted by the reputation system entries and it can choose to consider the ratings of other nodes or to rely solely on its own observations. One node can have varying reputation records with other nodes across the network, and the subjective view of each node determines its actions. Byzantine robustness [16] in the sense of being able to tolerate a number of erratically behaving servers or in this case nodes is the goal of a reputation system in mobile ad-hoc networks. Here, the detection of malicious nodes by means of the reputation systems has to be followed by a response in order to render these nodes harmless.

3 Solution Proposal: A Bayesian Approach to Reputation Systems

In this section we give details on how in our system reputation ratings are built and updated, how the ratings of other nodes are considered, and how decisions about future cooperation are made. For an overview of the Bayesian concepts used we refer the interested reader to the appendix.

3.1 Rating Representation

We propose to use a Bayesian approach for the representation and building of reputation as well as for subsequent decision-making depending on the reputation. Since the true probability of a node to act maliciously, say θ , is unknown, we make an estimation of θ by inference from the data X obtained by direct or indirect observations.

A binomial likelihood is assumed as $P(X) = \theta^n(1 - \theta)^{1-n}$. The process of updating ratings is as follows. First, choose a prior. To represent a non-informative prior and thus a uniform likelihood, we use $\text{Beta}(1, 1)$. Then calculate the posterior distribution and update at each observation. We use s to represent the number of successes and f for the number of failures. Then, $\text{Beta}(\alpha, \beta)' = \text{Beta}(\alpha', \beta')$ with $\alpha' = \alpha + s$ and $\beta' = \beta + f$.

The advantage of using the Beta function is that it only needs two parameters α and β that are continuously updated as observations are made or reported. These two parameters reflect the current rating, the higher the Beta curve, the more evidence samples have been taken in. The higher the peak and the narrower, the higher the confidence in the rating that there is a certain probability around which the observations center. Figure 1(a) shows the non-informative flat prior of $\text{Beta}(1, 1)$, all probabilities of θ are equally likely. After some updates according to observations of successes and failures, the posterior density

is depicted in Figure 1(d). The actual calculation of the density has been carried out here for illustrative purposes.

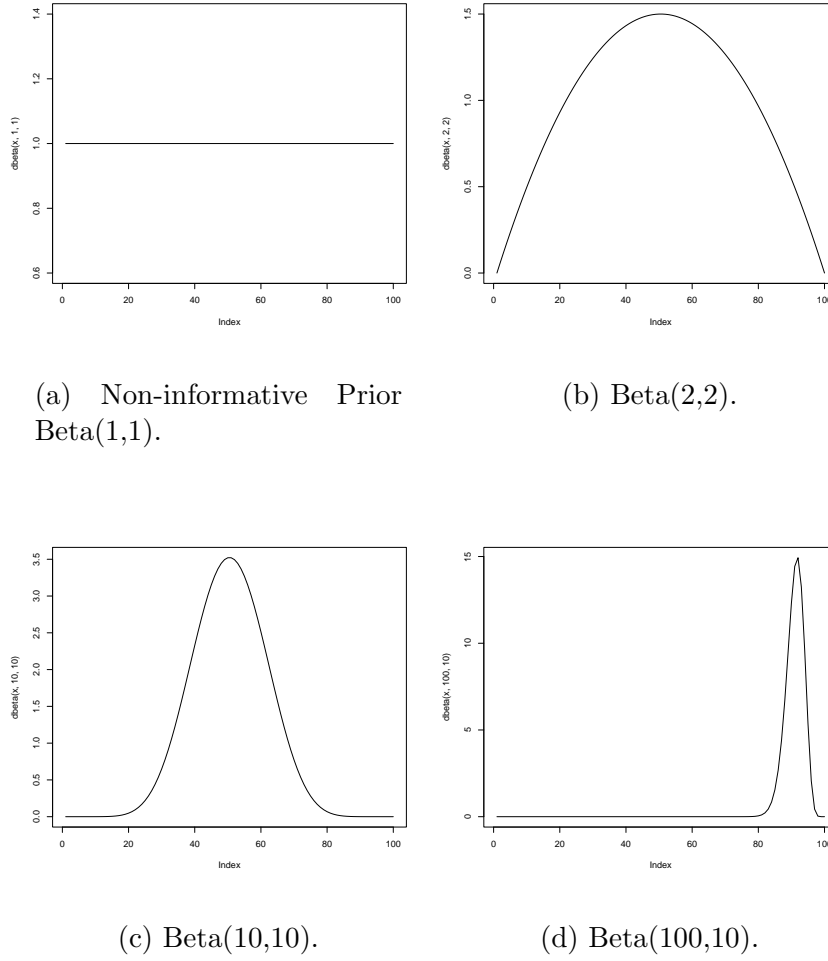


Fig. 1. Density of the Beta Function of Various Observations.

Applied to a reputation system, every node, say i , has a reputation component that receives as input first- or second-hand behavior information on other nodes, say j . It outputs decisions (misbehaving or not) for those j s where node i feels able to say something. We call $R_{i,j}$ the summarized data that captures j 's rating by i . $R_{i,j}$ is modified as information are received according to the update of the Beta function as explained above.

3.2 Merging Models

Before merging, we choose the models we deem likely to be correct. The question is how to detect and avoid false accusations. Our approach is to exclude those $R_{k,j}$ for which there is a large incompatibility between $R_{k,j}$ and $R_{i,j}$ for

some j . As a simple means to express that $R_{k,j}$ makes a strong case that j is malicious, whereas $R_{i,j}$ does not, we exclude $R_{k,j}$ from the model merging if it deviates from $R_{i,j}$ by more than d , the deviation threshold, in either direction.

Once we have the compatible ratings, we weight them before merging. In a first approach, we give the most weight to each nodes' own observations and we do not assume any a priori knowledge on the trustworthiness or expertise of a node. We thus weight second-hand information equally among the witnesses (e.g. neighbors) that share their ratings. The weight for each rating at an exchange encounter is thus $\frac{1}{w}$, w being the number of witnesses at this particular instant.

We use a linear opinion pool for model merging. The updated rating is thus:

$$R'_{i,j} = R_{i,j} + \sum_{k=1}^w \frac{1}{w} R_{k,j} \quad (1)$$

As an alternative approach we could conceive of modeling the trust given to particular nodes and have the respective weight depend on it. A trust component qualifies the trust that node i puts on second-hand information originated by other nodes, say k . We call $T_{i,k}$ the summarized data that captures the trust that node i places on node k . $T_{i,k}$ could first be configured by an external mechanism or be adaptive to the behavior in information dissemination, taking into account the accuracy. When the reputation system receives second-hand information (from k , about j), it would then use $T_{i,k}$ to decide how to update $R_{i,j}$ and to determine whom to send ratings to. As mentioned in Section 1, we chose not to use trust.

3.3 Decision Making

The decision-making process works as follows. First, the posterior according to all the given data is calculated. This we do by updating $R_{i,j}$ as explained above. Then we choose the decision with minimal loss.

We use squared-error loss for the deviation from the true θ , which is minimized by choosing $\mathbb{E}(\text{Beta}(\alpha, \beta))$. We now choose a decision $\delta^*(X)$ out of $\delta_1(X)$, classifying node j as regular, and $\delta_2(X)$, classifying node j as malicious. The decision $\delta^*(X)$ of choice is determined by the threshold t in Equation 2.

$$\delta(X)^* = \begin{cases} \delta_1(X) & \text{if } \mathbb{E}(\text{Beta}(\alpha, \beta)) < t \\ \delta_2(X) & \text{if } \mathbb{E}(\text{Beta}(\alpha, \beta)) \geq t \end{cases} \quad (2)$$

4 Performance Evaluation

4.1 Goals and Metrics

By means of simulation, we want to investigate the robustness and efficiency of a distributed reputation system in a mobile ad-hoc network. The key questions addressed are

- How long does it take until a malicious node is detected, using first-hand information only, using also second-hand information, i.e., the first-hand information of others, or even more indirect disseminated information?
- What is the effect of false accusations and can they be detected?
- How robust is the system to wrong observations?
- With whom should information be exchanged – with neighbors or remote nodes?
- And, what is the effect of mobility?

4.2 Simulation Setup

The simulation was implemented in R [9, 19]. To simulate good and malicious behavior, neighborhood, observation mistakes, movement, and reputation updates, we used a grid of nodes. We investigated and compared the effect of using first-hand information only, using also second-hand information in a network with no false accusations, and using also second-hand information in a network with liars but discarding too deviant ratings.

4.3 System Model

The nodes are placed on a grid, to simulate a communications range of one hop, and they observe the behavior of their neighborhood. Depending on its position in the grid, a node has up to 8 neighbors. A node can only directly observe neighbors, i.e., node i at row j and column k , denoted as i_{jk} , can observe any neighboring node n in its row $n_{j, <k+1|k-1>}$, in its column $n_{<j+1|j-1>, k}$, or diagonally one hop away $n_{<j+1|j-1>, <k+1|k-1>}$.

Periodically, nodes move around. We emulate this with the following algorithms.

Local movement. We pick a node at random, say node $i_{j,k}$ and randomly select a new location (j', k') for it such that $j' = [j - 2, j + 2]$ and $k' =$

$[k - 2, k + 2]$ to keep the movement reasonably local. We then repeat this with the node that we find at (j', k') and so on, until the new location is the original (j, k) and the permutation cycle is completed.

Local plus far movement. Most of the time the nodes move within a two-hop radius as described above, but sporadically they choose a location with long-distance hops.

Random movement. With this movement model, the new position of the nodes is a random permutation of the previous position.

Before moving away, nodes exchange reputation information in the form of Beta parameters. We have different models for the choice of witnesses.

Neighbors. Nodes exchange their reputation information with all nodes that are reachable within one hop. This way, the information dissemination does not need routing nor uses resources across the network.

A random set of nodes. Nodes pick their witnesses at random, so the information does not only spread locally but to wherever the chosen nodes are located at the moment of exchange. In a mobile ad-hoc network this model would consume more network resources than the neighbor model.

Friends. Again, the choice of witnesses is independent of location, but this time it is always the same set of nodes used to exchange ratings.

At each exchange the nodes give their ratings the way they stored their first-hand information. Liars apply different strategies to give false ratings, as explained in Section 1. We thus have the following liar models.

Reverse. When a node k lies, it swaps the α and β of its $\text{Beta}_{k,j}(\alpha, \beta)$ for all nodes j represented by $R_{k,j}$ before disclosing it to the neighbors for model comparison.

Worsen. Liars increase α of benign nodes by 20%.

Improve. Liars increase β of malicious nodes by 20%.

This whole process of observing, exchanging ratings, and moving is iterated until all of the malicious nodes are classified as detected by all of the nodes in the network, which is the case when the expected value of the reputation, $\mathbb{E}(\theta)$ represented by $\mathbb{E}(R_{i,j})$, exceeds a threshold of 0.75. As a rehabilitation mechanism to mitigate the effect of false accusations, the nodes periodically review their reputation ratings and reverse their classification from “detected” to “regular” when the reputation is substantially better than the detection threshold.

The threshold used to determine when to exclude a suspect liar’s rating depends on the priorities. As is typical for diagnosis systems, there is a trade-off between minimizing false positives or false negatives. We chose a threshold of 50% deviation to err on the side of false positives, i.e., the mechanism excludes some true information but reliably prevents false accusations from

having an impact. This way the robustness is maintained at the price of an unused detection speed-up potential.

4.3.1 Scenarios

We evaluate six scenarios that differ in whether disseminated information is considered at all, what kind of disseminated information is considered, and how it is integrated in the rating of a node. The following is a list of these scenarios with their names as they are used in the simulation.

First-hand information. $n_t(i)$ denotes the nodes that node i can observe during the time interval t , i.e. the grid neighbors. Each node j issues a sequence of bits out of $[0, 1]$ according to a distribution that depends on whether a node is benign, using $\mathbb{P}(\text{output}_{\text{benign}})$, or malicious, $\mathbb{P}(\text{output}_{\text{malicious}})$. Node i sees the bits correctly with $\mathbb{P}(\text{correctObservation})$.

- (1) Place nodes in the grid.
 - (2) \forall nodes, select $\text{type} \in \{\text{benign}, \text{malicious}\}$ and according probability distribution of output $\mathbb{P}(\text{output}_{\text{type}})$.
 - (3) repeat
 - (a) \forall nodes output byte according to $\mathbb{P}(\text{output}_{\text{type}})$.
 - (b) \forall nodes i , observe neighbors n correctly with probability $\mathbb{P}(\text{correctObservation})$.
 - (c) \forall nodes i , n update $R_{i,n}$ using the Beta function.
 - (4) until $t > o$, o being the number of observations at each location.
 - (5) Pick node, move until cycle completed. Repeat 1–3.
- until end of simulation, then \forall nodes i and j evaluate $R_{i,j}$ and compare to the type_j .

Second-hand information. (1) Iterations of the algorithm above.

- (2) Before moving, \forall nodes i and j output $R_{i,j}$.
- (3) \forall nodes i and j update $R_{i,j}$ by integrating local $R_{i,j}$ and $R_{k,j}$, the exchange partners' $R_{i,j}$.

Deltas only. Same as second-hand information, but use only the delta between the $R_{k,j}$ received at the last encounter and the current $R_{k,j}$.

Third-hand information. Nodes do not only exchange their respective first-hand information, but their second-hand information. Third-hand information is not independent but reinforcing beliefs by potentially mirroring them back to the originator, hence we only show the scenario for comparison.

With lies. Contaminated second-hand information.

We use probability distributions $\mathbb{P}(\text{tellTruth}_{\text{honest}})$ (probability of telling the truth as an honest node) and $\mathbb{P}(\text{tellTruth}_{\text{liar}})$ (probability of telling the truth when a node is a liar). Independent of its status as a benign or malicious type, nodes can be liars or honest.

- (1) Iterations of second-hand algorithm, but drawing from the probability distribution to tell a lie or the truth.

- (2) Compare $R_{i,j}$ with all witnesses k , weight $R_{k,j}$ by $\frac{1}{w}$, w being the number of witnesses considered, and integrate with $R_{i,j}$.
- (3) Include the contaminated information regardless.

Lies excluded. When comparing, only use $R_{k,j}$ s according to the compatibility metric, deviating less than d from $R_{i,j}$, with d being the deviation threshold and $R_{i,j}$ the accumulated reputation of j as seen by node i .

4.4 Factors and Parameters

In Table 1 we list the factors varied throughout the simulation, Table 2 contains the unchanged parameters.

Factor	Level 1	Level 2	Level 3
Number of nodes	25	49	100
$\mathbb{P}(\text{being malicious})$	0.1	0.5	0.9
$\mathbb{P}(\text{being a liar})$	0.1	0.5	0.9
Witnesses	neighbors	friends	random set
Liar strategy	reverse	worsen	improve
Mobility	local	local plus far	random

Table 1
Factors and their Levels

Parameter	Level
observations before movement	10
$\mathbb{P}(\text{output}_{benign})$	0.99
$\mathbb{P}(\text{output}_{malicious})$	0.99
$\mathbb{P}(\text{correctObservation})$	0.99
$\mathbb{P}(\text{tellTruth}_{honest})$	0.99
$\mathbb{P}(\text{tellTruth}_{liar})$	0.99
t , the threshold for detection	0.75
d , the deviation threshold	0.5

Table 2
Fixed Parameters

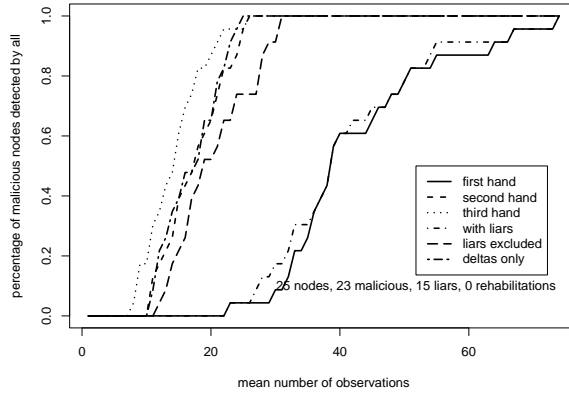


Fig. 2. Mean Detection Time of All Malicious Nodes by All 25 Nodes.

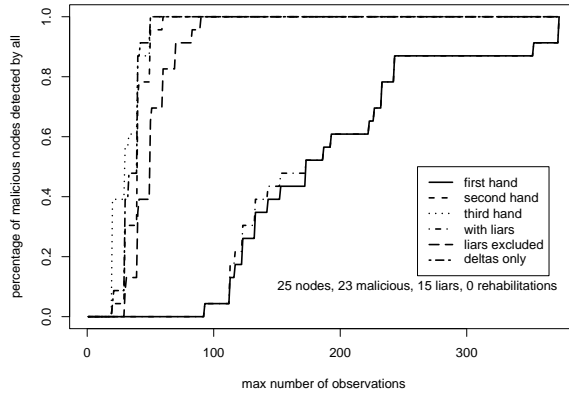


Fig. 3. Max Detection Time of All Malicious Nodes by All 25 Nodes.

4.5 Results

Figure 2 shows the mean detection time, i.e., the time in the simulation when the last node detected a particular malicious node, vs. which fraction of the malicious nodes were detected by all at that time, Figure 3 shows the maximum detection time for all nodes. Figures 4 and 5 show examples of larger networks, also varying the number of malicious nodes and the number of liars. These examples are representative of the results obtained by the simulation. We chose to show individual representative examples for this type of plot of detection fraction versus time instead of mean outcomes over several runs, since the type of a node both concerning the cooperation and the lying properties are drawn from probability distributions and not explicitly specified, thus the portion of malicious nodes or liars varies. However, for the mean of the mean detection time by all nodes and the maximum of the max detection time by all nodes, we consider several simulation runs in Figure 6.

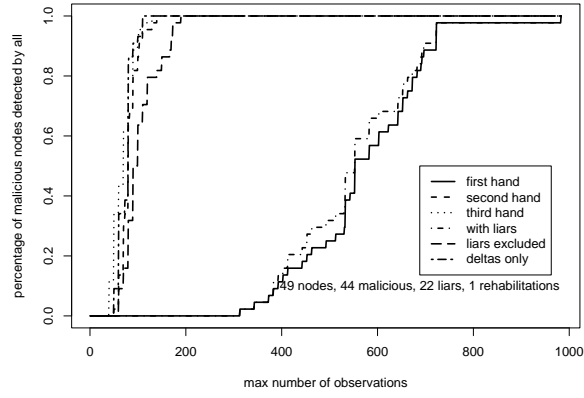


Fig. 4. Max Detection Time of All Malicious Nodes by All 49 Nodes.

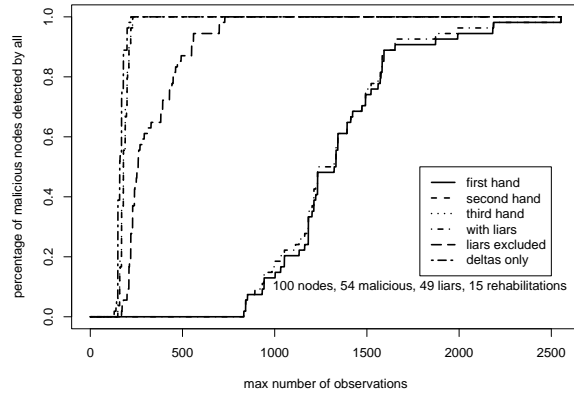


Fig. 5. Max Detection Time of All Malicious Nodes by All 100 Nodes.

Using the full set of second-hand information or using only the difference between already received second-hand information and the current second-hand information consistently perform very similarly and very well. Exchanging the full set of observations when nodes encounter repeatedly considers information as new that has been integrated already and thus can bias the belief, whereas keeping track of the last exchanged information, albeit only two parameters per reputation, can add up to a significant storage requirement in large mobile networks.

Over the course of the simulation, it has emerged that using the ‘liars excluded’ Bayesian scenario significantly improves on the performance of the mean detection time when compared to the ‘first hand’ scenario, yet the performance gain is even higher in the worst case, namely the maximum detection time, i.e., the maximum time it takes for a malicious node to be deemed ‘detected’ by all the nodes of the network.

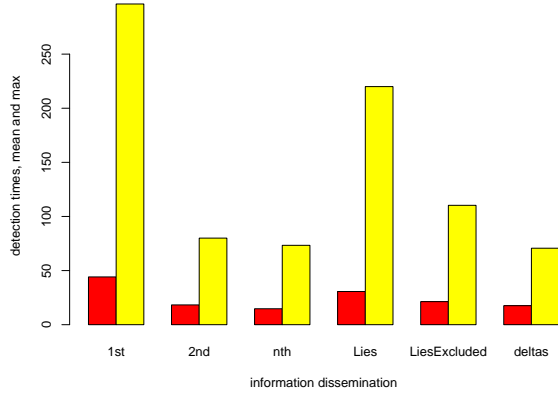


Fig. 6. Mean and Max Detection Timex vs. Information Dissemination.

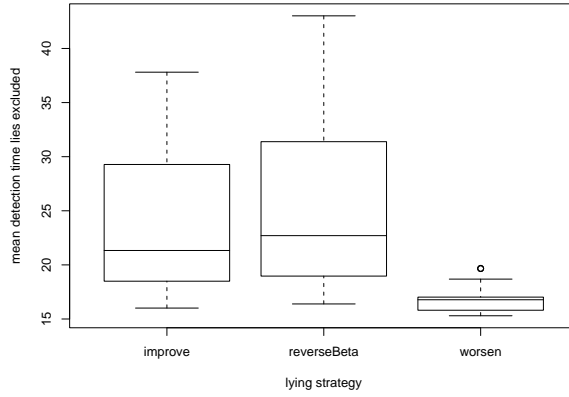


Fig. 7. Mean Detection Time (Lies Excluded) vs. Lying Strategy.

Another observation is that, as one would expect, the detection improvement given by the use of second-hand information even in the presence of liars, but given the attempt to discard the false accusations by means of our Bayesian approach, in fact increases with the network size. The larger the network, the higher the probability of receiving information about nodes before actually encountering them as neighbors and being able to observe their behavior.

When nodes not only exchange their own first-hand information but hand on disseminated information of a deeper transitivity level, their own ratings once voiced can be reflected to them at a later time, thus reinforcing their original rating. Although using this 'third-hand' or 'nth-hand' information consistently outperforms all other strategies, it is not a valid choice since these ratings are not independent.

For networks of 25 nodes, some effects of varying the level of the factor of the lying strategy are shown in Figures 7, 10, and 11. The mobility impact is

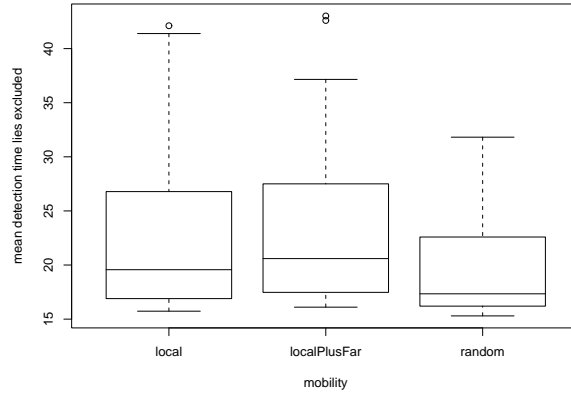


Fig. 8. Mean Detection Time (Lies Excluded) vs. Mobility.

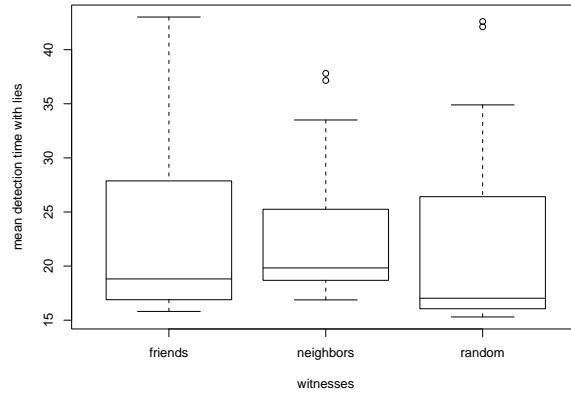


Fig. 9. Mean Detection Time (Lies Excluded) vs. Witnesses.

shown in Figures 8, 12, 13, and the choice of witnesses is depicted in Figures 9 and 14. Except for the mobility factor, none of the others had an impact on either the first-hand information or the truthful second-hand information scenario.

As can be seen from Figure 15, the performance of the Bayesian approach of liar exclusion improves when the number of liars is small and approaches the performance of truthful second-hand information. In the presence of many liars, the performance degrades gradually but is still better than relying only on first-hand information. In all the figures, the scenario ‘with lies’, i.e., integrating contaminated second-hand information regardless, performs better than relying on first-hand information only, yet the price for this speed-up in detection time is that innocent nodes are also being classified as ‘detected’ by many nodes due to the effect of false accusations. This has consistently been avoided by the ‘liars excluded’ scenarios throughout the entire simulation.

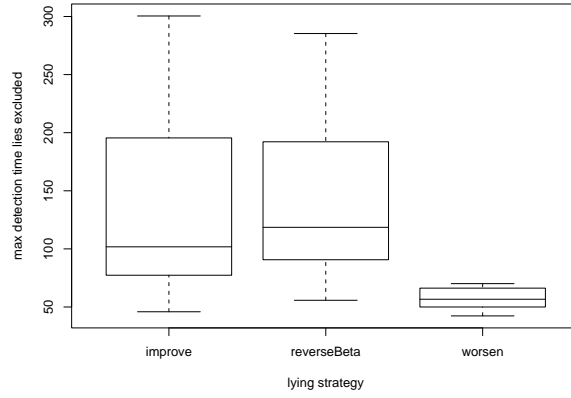


Fig. 10. Max Detection Time (Lies Excluded) vs. Lying Strategy.

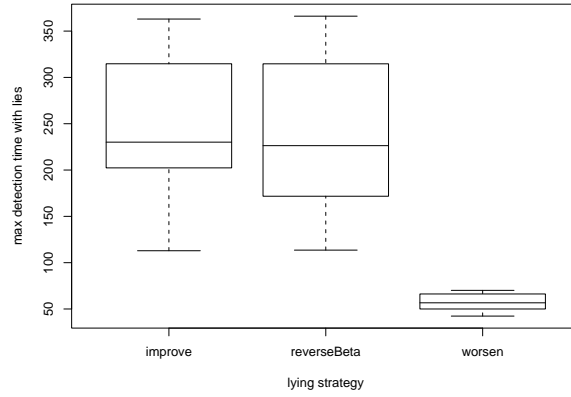


Fig. 11. Max Detection Time (With Lies) vs. Lying Strategy.

Figure 16 shows that only the 'reverse' lying strategy led to effective false accusations, i.e. false accusations that lead to the classification of benign nodes as malicious. The number of effective false accusations increased with a growing population of liars. The impact of false accusations was largely reduced by the Bayesian approach (Lies Excluded).

5 Discussion/Future Work

We are working on an extending the CONFIDANT protocol and its simulation implementation with the Bayesian approach we proposed in this paper. This way we can incorporate the insights gained to make the protocol robust against false accusations yet reasonably fast in detection. It will also enable us to evaluate our Bayesian approach in a more realistic mobile ad-hoc network. We

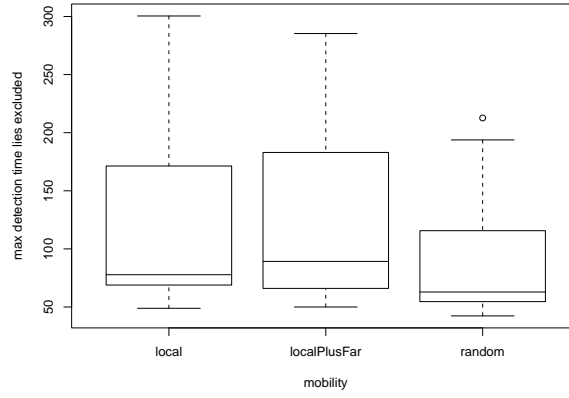


Fig. 12. Max Detection Time (Lies Excluded) vs. Mobility.

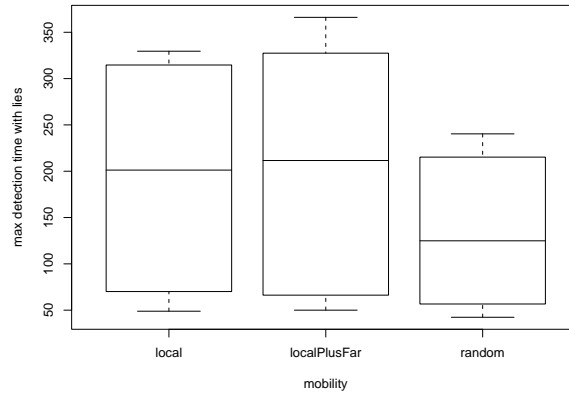


Fig. 13. Max Detection Time (With Lies) vs. Mobility.

then aim for more simulation runs and exploring a larger space of parameters in this environment.

For instance, the current system is dampening the effect of false accusations by forcing liars to lie more to have a fast effect, but then detection is easier. To combat the slow deliberate degradation of reputation, we intend to introduce an aging mechanism of reputation into the simulation.

Also, a side-effect of the emphasis on robustness was that, given the nature of a node did not change throughout the simulation time, the rehabilitation mechanism provided for the strategy of excluding liars was rarely required. This might not be the case with more elaborate adversary models that we intend to consider.

We have not solved all of the following challenges for merging ratings.

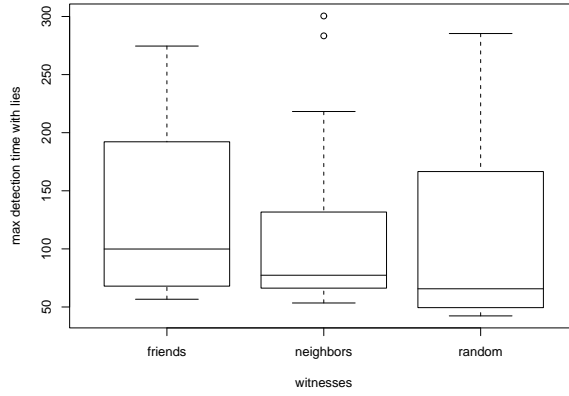


Fig. 14. Max Detection Time (Lies Excluded) vs. Witnesses.

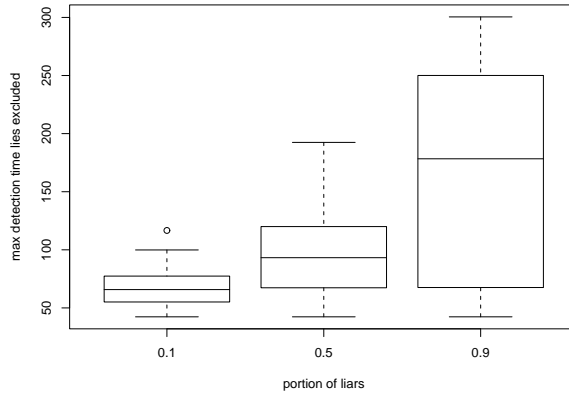
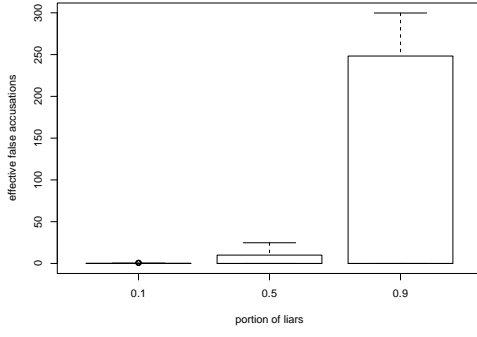


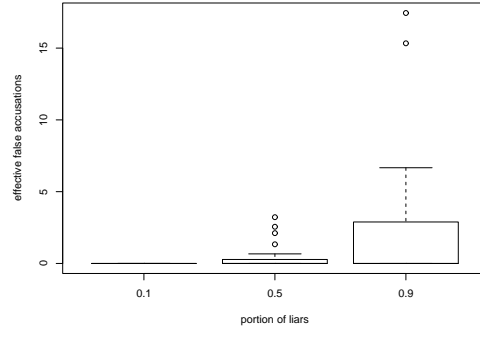
Fig. 15. Max Detection Time (Lies Excluded) vs. Portion of Liars.

- False/fake ratings for deliberate deception and influence.
- Contradicting models. How to consolidate them, whom to believe, how to assign weights for significance.
- Privacy concerns. Nodes may not want to expose their ratings to others, also there is a reduction of uncertainty which might be beneficial to malicious nodes.
- With whom to share information. Who provides the most valuable information, who is trusted for their rating, and, related to the privacy concern, whom can nodes show their ratings without harm.

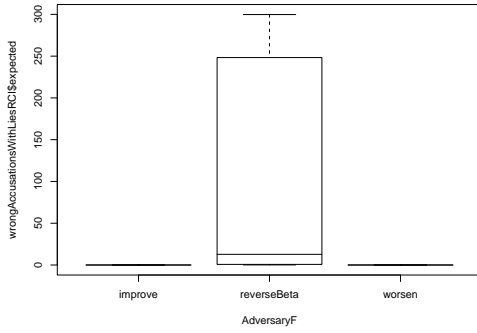
Evaluating our approach within the CONFIDANT protocol implementation will also enable us to see the overhead involved with the choice of witnesses in terms of routing. Furthermore, observations might not be so clearly classifiable in more realistic mobile ad-hoc environments due to collisions, link-layer errors, and asynchronous moving of nodes. We can thus better evaluate the impact of incorrect observations.



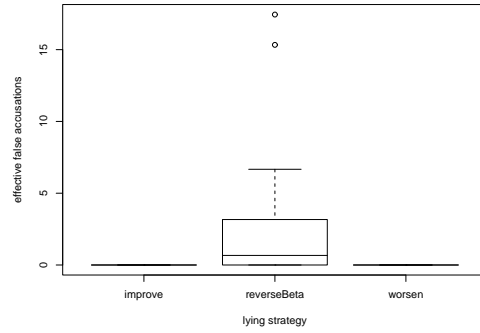
(a) Effective False Accusations vs. Lying Strategy (With Lies).



(b) Effective False Accusations vs. Lying Strategy (Lies Excluded).



(c) Effective False Accusations vs. Portion of Liars (With Lies).



(d) Effective False Accusations vs. Lying Strategy (Lies Excluded).

Fig. 16. Effective False Accusations.

6 Conclusions

Using second-hand information can significantly accelerate the detection and subsequent isolation of malicious nodes in mobile ad-hoc networks. However, if nodes are deceived by wrong observations or false accusations, the robustness of the reputation system is endangered.

In this paper we presented and evaluated a Bayesian approach for reputation representation, integrating disseminated information, and coping with false accusations. We found that, enabled by our Bayesian approach, by excluding ratings that deviate substantially from first-hand information and the majority rating of second-hand ratings gathered over time, robustness of the reputation system against false accusations is largely achieved. This holds true even with

a large number of liars in the network. As opposed to relying exclusively on first-hand information, the increased robustness of our approach does not have to be traded off against longer detection delays. The detection speed improves significantly over merely using first-hand information and, with a decreasing portion of liars, approximates the ideal case of using truthful second-hand information.

References

- [1] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2001)*, 2001.
- [2] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition edition, 1985.
- [3] Sonja Buchegger and Jean-Yves Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403 – 410, Canary Islands, Spain, January 2002. IEEE Computer Society.
- [4] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002. IEEE.
- [5] Anthony Davison. *Bayesian Models*. Chapter 11 in Manuscript, 2002.
- [6] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, Inc., 1970.
- [7] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 150–157, 2000.
- [8] D. Hoeting, J. A. Madigan and C.T. Raftery, A.E. and Volinsky. Bayesian model averaging: A tutorial (with discussion). *Statistical Science*, 44(4):382–417, 1999.
- [9] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.
- [10] Peter Kollock. The production of trust in online markets. *Advances in Group Processes*, edited by E. J. Lawler, M. Macy, S. Thyne, and H. A. Walker, 16, 1999.
- [11] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.
- [12] Pietro Michiardi and Refik Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks.

- Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia., 2002.
- [13] Pietro Michiardi and Refik Molva. Simulation-based analysis of security exposures in mobile ad hoc networks. European Wireless Conference, 2002.
 - [14] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifiable(sucv) identifiers and addresses. NDSS'02, February 2002., 2002.
 - [15] Krishna Paul and Dirk Westhoff. Context aware inferencing to rate a selfish node in dsr based ad-hoc networks. In *Proceedings of the IEEE Globecom Conference*, Taipeh, Taiwan, 2002. IEEE.
 - [16] Radia Perlman. Network layer protocols with byzantine robustness. PhD. Thesis Massachusetts Institute of Technology, 1988.
 - [17] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. Working Paper for the NBER workshop on empirical studies of electronic commerce, 2001.
 - [18] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.
 - [19] William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S-Plus. Third Edition*. Springer, 1999. ISBN 0-387-98825-4.

A APPENDIX: Relevant Bayesian Background

A.1 Belief Representation Using the Beta Function

Bayes' Theorem is shown in Equation A.1. It is used to calculate the probability of a random variable given an observation.

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_{i=1}^n P(A|B_i)P(B_i)} \quad (\text{A.1})$$

A prior distribution (prior to receiving information) reflects the initial belief. Any up-front information can be fed into the prior to give it a head start. The prior, however, can also be chosen such that it reflects ignorance or indifference toward the initial situation. Given this prior, at each observation the information available is updated to reflect the added knowledge and to increase the precision of a belief. If the likelihood of a property is binomial, i.e., successes and failures occur independently, a good prior density is the Beta function. The Beta function is the conjugate prior for binomial likelihood and thus the posterior (after taking into account the received information) density is also

Beta [2, 6]. The Beta function is used to reflect the prior belief. It is defined as follows.

$$f(\theta) = \text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \quad (\text{A.2})$$

$$\Gamma(x + 1) = x\Gamma(x), \Gamma(1) = 1 \quad (\text{A.3})$$

The posterior is given and updated at each observation in the following way. We use s to represent the number of successes and f for the number of failures. Then, $\text{Beta}(\alpha, \beta)' = \text{Beta}(\alpha', \beta')$ with $\alpha' = \alpha + s$ and $\beta' = \beta + f$.

The Beta function offers moments that are simple to calculate.

$$\mathbb{E}(\text{Beta}(\alpha, \beta)) = \frac{\alpha}{\alpha + \beta} \quad (\text{A.4})$$

$$\sigma^2(\text{Beta}(\alpha, \beta)) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)} \quad (\text{A.5})$$

A.2 Model Merging

In their tutorial on Bayesian model averaging, Hoeting et al. [8] give the following methodology.

If Δ is the quantity of interest, such as an effect size, a future observable, or the utility of a course of action, then its posterior distribution given data D is:

$$\mathbb{P}(\Delta|D) = \sum_{k=1}^K \mathbb{P}(\Delta|M_k, D)\mathbb{P}(M_k|D). \quad (\text{A.6})$$

This is an average of the posterior distributions under each of the models considered, weighted by their posterior model probability. M_i, \dots, M_k are the models considered. The posterior probability for model M_k is given by

$$\mathbb{P}(M_k|D) = \frac{\mathbb{P}(D|M_k)\mathbb{P}(M_k)}{\sum_{l=1}^K \mathbb{P}(D|M_l)\mathbb{P}(M_l)} \quad (\text{A.7})$$

where

$$\mathbb{P}(D|M_k) = \int \mathbb{P}(D|\theta_k, M_k)\mathbb{P}(\theta_k|M_k)d\theta_k \quad (\text{A.8})$$

is the integrated likelihood of model M_k , θ_k is the vector of parameters of model M_k , $\mathbb{P}(\theta_k|M_k)$ is the prior density of the parameters under model M_k , $\mathbb{P}(D|\theta_k, M_k)$ is the likelihood, and $\mathbb{P}(M_k)$ is the prior probability that M_k is the true model. All probabilities are implicitly conditional on \mathcal{M} , the set of all models considered.

In addition, Davison [5] lists the following, with z being the variable of interest, and y the data.

$$f(z|M_i, y) = \frac{\int f(z|y, \theta_i, M_i) f(y|\theta_i, M_i) \pi(\theta_i|M_i) d\theta_i}{f(y|M_i)} \quad (\text{A.9})$$

Here θ_i is the parameter for model M_i , under which the prior is $\pi(\theta_i|M_i)$ and the prior probability of M_i is $\mathbb{P}(M_i)$.

Berger [2] lists several methods for combining probabilistic evidence. To process different sources of information, he lists two ad-hoc systems.

Linear Opinion Pool. Assign a positive weight w_i (where $\sum_{i=1}^m w_i = 1$) to each information source π_i (supposedly to reflect the confidence in that information source), and then use

$$\pi(\theta) = \sum_{i=1}^m w_i \pi_i(\theta) \quad (\text{A.10})$$

Independent Opinion Pool. When the information sources seem “independent”, use, as the overall probability distributions for θ ,

$$\pi(\theta) = k \left[\prod_{i=1}^m \pi_i(\theta) \right] \quad (\text{A.11})$$

The alternative to the use of ad-hoc rules is, according to Berger, probabilistic modeling, i.e., obtaining the joint distribution of all random observables and unknown parameters of interest or, at least, determining enough to calculate the conditional (posterior) distribution of the desired θ given the observables. This is sometimes called the *super Bayesian* approach, to emphasize that it is a single decision maker (the super Bayesian) who is trying to process all the information to arrive at a distribution of θ which is consistent with probabilistic reasoning.

$$\pi(\theta_1|p) = \left[1 + \frac{(1-p)^{\alpha-\beta}}{p} \frac{\pi_2(\theta_2)}{\pi_2(\theta_1)} \right]^{-1} \quad (\text{A.12})$$

The goal is to minimize risk. Loss can be represented as squared-error loss or 0-1 loss for classification, for instance, as depicted in equations A.13 and A.14.

$$L(\theta, \alpha) = (\theta - \alpha)^2 \tag{A.13}$$

$$L(\theta, \alpha_i) = \begin{cases} 0 & \text{if } \theta \in \Theta_i \\ 1 & \text{if } \theta \in \Theta_j, j \neq i \end{cases} \tag{A.14}$$

Then, for all actions the loss is calculated and weighted by its likelihood. Finally, the action δ^* with the smallest risk R (expected loss L) is chosen from $R(\theta, \delta^*) = \mathbb{E}[L(\theta, \delta(X))]$.